

Descripción del problema

Se presenta el problema de crear un sistema de consulta de recetas para un restaurante con el objetivo de agilizar el trabajo del chef, ya que ha tenido problemas para memorizar las recetas. El sistema debe tener la opción de trabajar en dos modalidades: mantenimiento y consulta.

En cuanto al diseño del sistema se presenta el problema de conectar Prolog con algún otro lenguaje, ya que el sistema se debe componer de dos partes: "Front-end y Back-end. Siendo Front-end el encargado de manejar la interacción con el usuario y debe estar escrito en otro lenguaje de programación.

El Back-end (Prolog) tiene la tarea de manejar la base de conocimientos de las recetas. Cada parte debe estar conectada para las tareas relacionadas con mantenimiento de bases y consulta.

Para esta tarea se decidió utilizar python ya que por la investigación su biblioteca de conexión pypi era más fácil manejar los predicados que en otro lenguaje.

Decisiones de diseño.

Respecto a este punto se decidió al final por usar un solo predicado llamado receta el cual contiene toda la información la estructura del mismo es la siguiente.

```
"receta(Nombre,Autor,Estilo,Ingrediente,Preparacion)"
#Estilo de receta describe si una receta es francesa, italiana o de otro estilo.
```

Respecto a cómo se manejan los predicados al momento de actualizar y borrar esto es un trabajo en archivos sobre "recetas.pl" y "recetas.txt" en recetas.pl se encuentran los predicados del sistema para cargar a prolog y el archivos se encuentran solo los nombres de las recetas para cuando deban eliminarse o actualizarse, el sistema busca la receta en el archivo .txt por el nombre apartir de eso busca el predicado y lo borra del archivo recetas/pl para despues volver a cargar los predicados

En lo que se refiere a búsqueda en el sistema arma el predicado:

```
"receta(Nombre,Autor,Estilo,Ingrediente,Preparación)"
```

e introduce la información ingresada por el usuario para hacer la consulta, si el usuario no coloca nada el sistema carga todas las recetas del sistema y las muestra en un campo de texto.

Para inserción y actualización se solicita colocar toda la información nuevamente esto por fiabilidad de la receta a modo de que se generaron cambios.

Para borrar una receta solo debe ingresar su nombre el sistema se encarga de borrarlo asi como se describió más arriba

Algoritmos usados

Este punto no será muy extenso ya que para su explicación detallada se tendrá el código fuente del programa.

Los algoritmos fueron:

Ventanas para la presentación de interfaz

Para metodo de sistema:

Verificadores de información que esta no fuera vacia
sistemas de armado de predicado a nivel de archivo
carga del archivo .pl para cargar los predicados
armado del predicado para la consulta
sistema de consulta para mostrar
elemento de texto para mostrar los resultados
sistemas de busqueda de recetas por indice y para obtener la información en caso de no existir
la receta tira error.

Diseño del programa

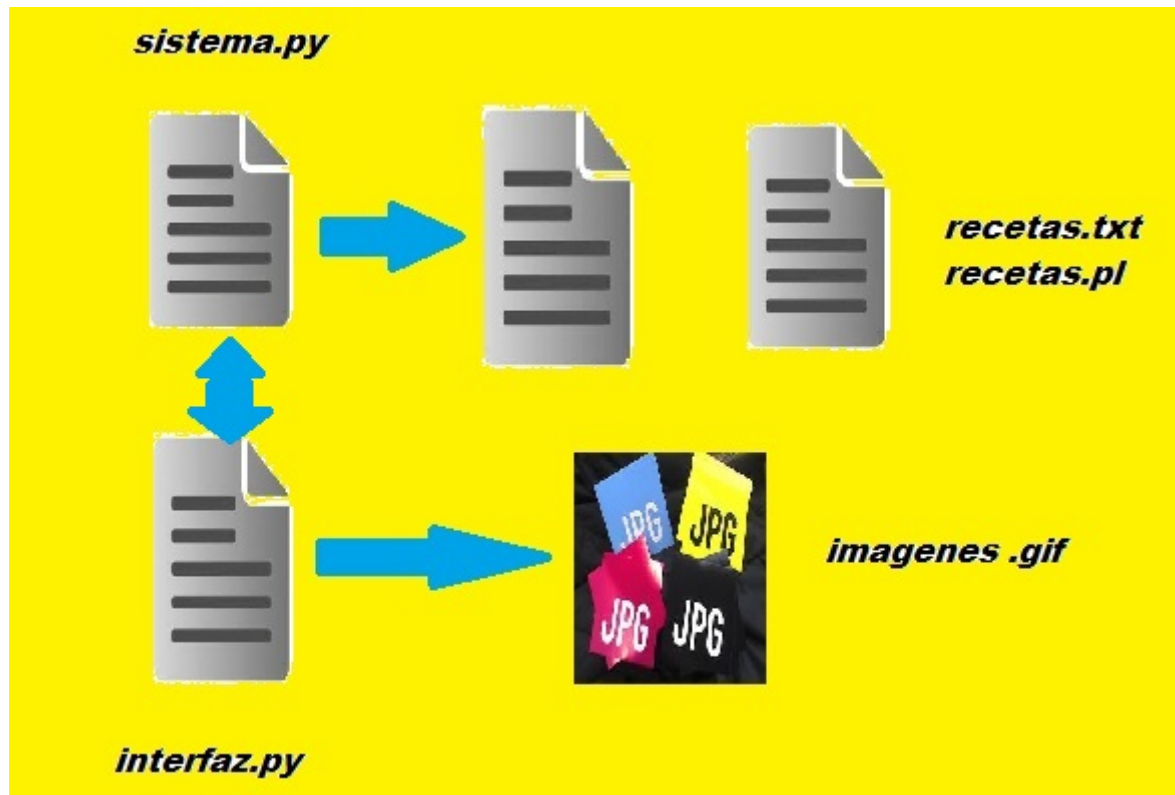
Para realizar el Front End de este programa, se escoge el lenguaje de programación Python, el cual es multiparadigma, interpretado y usa tipado dinámico, además de ser multiplataforma, características que ayudan a la elección del mismo. Además se presenta la ventaja, que todos los integrantes del grupo, ya alguna vez lo han utilizado, ahorrando tiempo de investigación. Para la interconexión entre Prolog (Back- end) y Python (Front-end) se usa swipl, que es un software libre, entorno prolog, licenciado bajo la Licencia Pública Menor GNU.

Además para facilidad de manejo se decidio hacer varios archivos.

recetas.pl #Manejo de los predicados de prolog
recetas.txt #Manejo de los nombre de recetas

interfas.py #Manejo de interfaz
sistema.py #Manejo de las funciones del programa

Representación del sistema



Librerías Usadas

from Tkinter import *	Se utiliza para realizar la interfaz del programa.
Import tkMessageBox	Se utiliza para enviar ventanas simples con mensajes para el usuario.
From string import *	Se utiliza para el manejo de strings.
Import os	Se utiliza para abrir el manual de usuario.
Import sys	Utilizado para salir del programa y cerrar la ventana principal.
From pyswip import *	Se utiliza para realizar la conexión de python-prolog.

Análisis de resultados

Se cumplieron los objetivos requeridos para la tarea programada:

- Inserción de nuevas recetas, con la excepción de que permite ingresar un solo ingrediente.
- Actualización y borrado de alguna receta.

- Consulta de una receta, ésta se puede hacer por medio del ingrediente, el nombre del chef, o del nombre de la receta o el estilo.
- Respecto al ingrediente no se logro hacer que el sistema pudiera controlar varios ingredientes. Para resolver el problema se piensa que se puede ingresar los ingredientes como lista, hacer una consulta normal y verificar al final que los ingredientes ingresados estan el la lista de ingredientes de la receta

Conclusión

Se cumplieron los objetivos establecidos al inicio del proyecto, se buscaron las mejores soluciones entre los miembros del grupo, buscando utilizar siempre la más adecuada. La integración entre Python y prolog, presentó algunas complicaciones, pero después de una investigación se logró detectar el inconveniente y se arregló para continuar. Se implementa la interfaz gráfica en python para presentar al usuario una interfaz agradable y fácil de manejar a la hora de incluir, modificar, borra o actualizar datos, permitiendo que cualquier Chef que tenga la necesidad de hacer uso de este programa pueda acceder sin ningún inconveniente.